Defence Research and Development Canada | Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

Staged Experiments in Mobile Vehicle Autonomy

Procedures and Results with the SegwayRMP

J. Giesbrecht, J. Collier, S. Monckton
Defence R&D Canada – Suffield

Technical Memorandum
DRDC Suffield TM 2004-288
December 2004

Canada

Report Documentation Page				Form Approved OMB No. 0704-0188
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.				
1. REPORT DATE **DEC 2004**		2. REPORT TYPE		3. DATES COVERED **-**
4. TITLE AND SUBTITLE **Staged Experiments in Mobile Vehicle Autonomy (U)**				5a. CONTRACT NUMBER
				5b. GRANT NUMBER
				5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)				5d. PROJECT NUMBER
				5e. TASK NUMBER
				5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Defence R&D Canada - Suffield,PO Box 4000, Station Main,Medicine Hat, AB,CA,T1A 8K6**				8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**				
13. SUPPLEMENTARY NOTES **The original document contains color images.**				
14. ABSTRACT **This paper presents details and results for experiments conducted at DRDC Suffield in autonomous robot control of the Segway RMP (Robotic Mobility Platform). The autonomous control was a demonstration of robotic abilities achievable with an inexpensive, rapid implementation by a small team using off-the-shelf hardware and software. It was based upon the Player/Stage robot software development environment, and used a basic sensor suite consisting of a SICK laser scanner, GPS localization, and robot distance encoders. The demonstration was able to achieve basic teleoperation, as well as autonomous deadreckoning, GPS waypoint following, obstacle avoidance and leader/follower behaviour. Safety procedures and robot performance metrics applicable to all autonomous vehicle research were also developed over the course of this project, and are documented in this report.**				
15. SUBJECT TERMS				
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES **54** / 19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Staged Experiments in Mobile Vehicle Autonomy

Procedures and Results with the SegwayRMP

J. Giesbrecht, J. Collier, S. Monckton
Defence R&D Canada – Suffield

Defence R&D Canada – Suffield

Technical Memorandum
DRDC Suffield TM 2004-288
December 2004

Author

J. Giesbrecht

Approved by

D. Hanna
Head/Tactical Vehicle Systems Section

Approved for release by

Dr. P. D'Agostino
Chairman/Document Review Panel

Abstract

This paper presents details and results for experiments conducted at DRDC Suffield in autonomous robot control of the Segway RMP (Robotic Mobility Platform). The autonomous control was a demonstration of robotic abilities achievable with an inexpensive, rapid implementation by a small team using off-the-shelf hardware and software. It was based upon the Player/Stage robot software development environment, and used a basic sensor suite consisting of a SICK laser scanner, GPS localization, and robot distance encoders. The demonstration was able to achieve basic teleoperation, as well as autonomous dead-reckoning, GPS waypoint following, obstacle avoidance and leader/follower behaviour. Safety procedures and robot performance metrics applicable to all autonomous vehicle research were also developed over the course of this project, and are documented in this report.

Résumé

Cet article présente les détails et les résultats des essais conduits à DRDC Suffield dans le domaine du contrôle des robots autonomes de Segway RMP (Plateforme sur la mobilité robotique). Le contrôle autonome a fait la démonstration des capacités robotiques réalisables avec une application rapide et économique qu’une petite équipe effectue au moyen de matériel et de logiciels ordinaires. Il est basé sur un milieu d’élaboration du logiciel pour robot Player/Stage et utilise des détecteurs et senseurs ordinaires qui consistent en un scanner au laser SICK, la localisation GPS et des robots encodeurs à distance. La démonstration a été en mesure de réussir la téléopération de base ainsi que la détection autonome à l’estime, le suivi d’un point de cheminement au GPS, l’évitement d’obstacle et un comportement de chef et d’exécutant. Des mesures de sécurité et la métrologie de la performance du robot applicables à toute la recherche sur les véhicules sans pilote ont aussi été mises au point au cours de ce projet et sont documentées dans ce rapport.

This page intentionally left blank.

Executive summary

Background: Through the Autonomous Land Systems (ALS) initiative, Defence R&D Canada – Suffield is investigating the area of Unmanned Ground Vehicles (UGVs). It is commonly assumed that providing robotic vehicles to assist military personal need be a complex, expensive undertaking. However, this is not necessarily the case. This project has successfully demonstrated several basic, but useful autonomous behaviours achieved with a small sensor suite and pre-existing software, and has shown that with a small amount of additional time and effort an even wider variety of relevant abilities could be developed. These behaviours were achieved with the Player/Stage open source software package, utilizing a TCP/IP radio network over Wave Wireless SpeedLan radios. Sensors included one SICK LMS200 laser scanner for obstacle detection, a Sokkia GMS2600 GPS unit for absolute position information, as well as the Segway platform's odometry sensors for dead-reckoning.

A secondary goal of the trials was to familiarize DRDC Suffield staff with the technical and safety problems of autonomous vehicle control and to demonstrate common approaches to robot navigation, with a goal of ramping the ALS team up in anticipation of a larger, far more complex demonstration in the fall of 2005.

Principle Results: With these goals in mind, the trials demonstrated and characterized the robot behaviour in a number of scenarios:

1. Vehicle Shutdown and Safety - Robot can be safely shut down and respond safely to a variety of hardware and software failures.

2. Teleoperation and Telemetry - Robot can be controlled remotely and can report back information such as laser scan readings, vehicle position, and robot state.

3. GPS Trajectory Following - Robot can be controlled in absolute position coordinates using GPS. Multiple way-points can be assigned to the robot for it to sequentially follow.

4. Dead-Reckoning - Robot can keep track of its position accurately.

5. Obstacle Avoidance - Robot can keep itself safe by avoiding both static and dynamic obstacles while making its way to goal locations.

6. Leader/Follower Behaviour - Robots are able to share information and work as a team while demonstrating all of the above characteristics.

7. Pursuit Behaviour - One robot can use information about another to pursue it in real-time.

In addition to these results, procedures for safely demonstrating these autonomous behaviours were created and a baseline for benchmarking future robotic development was established. A definition of an Autonomous Robotics Demonstration Team was also created, which consists of an Experiment Safety Officer, an Experiment Lead, Technical Support Personnel, Area Marshals, a Camera Operator and a Data Collection Officer.

Significance of Results and Future Work: Because of the successful use of Player/Stage in this project, off-the-shelf opens source software technology will continue to be be leveraged by DRDC in the future. In addition, a number of lessons were learned regarding software architecture, such as the importance of modularity and network centered design.

Furthermore, lessons were learned regarding the difficulties in making field ready robotic units. These robots were lacking in stability, and need more complex sensing, obstacle avoidance, and global path planning to function in more realistic scenarios. Efforts would also need to be directed at making them more rugged and less susceptible to environmental conditions.

Finally, these experiments proved that basic cooperative behaviours can be achieved fairly easily by communicating GPS positions through a radio network. Although these are important capabilities, reliance on GPS and explicit radio communications for cooperation is somewhat undesirable as they are subject to jamming and intermittent availability. In the future, cooperative behaviours should be based on visual recognition and relative position finding, with the use of as little bandwidth as possible. This will increase the robustness of teams of UGVs.

Giesbrecht, J., Collier, J., Monckton, S. 2004. Staged Experiments in Mobile Vehicle Autonomy. DRDC Suffield TM 2004-288. Defence R&D Canada – Suffield.

Sommaire

Contexte : R & D pour la défense, Canada – Suffield examine le domaine des Véhicules terrestres sans pilote, dans le contexte de l'Initiative des systèmes terrestres autonomes (STA). Généralement, on estime que fournir des véhicules robotisés au personnel militaire signifie une entreprise complexe et coûteuse. Cependant, ce n'est pas toujours le cas. Le projet a réussi à démontrer plusieurs comportements autonomes de base très utiles, réalisés avec des détecteurs et senseurs et un logiciel préexistant. Il a aussi démontré qu'avec un peu de temps et d'effort supplémentaire, il est possible de mettre au point une plus grande variété de capacités applicables. Ces comportements ont été réalisés avec le progiciel Player/Stage de source non secrète, en utilisant un réseau de radio TCP/IP avec des radios Wave Wireless SpeedLan. Les capteurs comprenaient un scanneur au laser SICK LMS200 pour la détection des obstacles, une unité GPS Sokkia GMS2600 pour l'information au sujet de la position absolue ainsi que les capteurs d'odométrie de la plateforme Segway pour la détection à l'estime.

L'objectif secondaire des essais était de familiariser le personnel de RDDC Suffield avec les problèmes techniques et de sécurité du contrôle autonome des véhicules et de démontrer des méthodes usuelles de la navigation robotisée, tout en ayant pour but de préparer l'équipe STA pour une démonstration beaucoup plus importante et plus complexe, prévue pour l'automne 2005.

Résultats principaux : Avec ces objectifs en vue, les essais ont démontré et caractérisé le comportement des robots dans un certain nombre de scenarios:

1. Arrêter et sécuriser un véhicule - Les robots peuvent être arrêtés en toute sécurité et répondre à une variété de pannes de matériel et de logiciels.

2. Téléopération et télémétrie - Un robot peut être contrôlé à distance et peut rapporter l'information telle que la lecture du scanneur au laser, la position du véhicule et l'état du robot.

3. Suivre la trajectoire GPS - Un robot peut être contrôlé avec les coordonnées de la position absolue en utilisant un GPS. Plusieurs points de cheminement peuvent être assignés au robot lui permettant de suivre le chemin en séquence.

4. Détection à l'estime - Un robot peut retrouver sa position avec précision.

5. Évitement des obstacles - Un robot peut se rendre à sa nouvelle position tout en restant sécuritaire et en évitant les obstacles statiques et dynamiques.

6. Le comportement de chef et de l'exécutant - les robots sont capables de mettre l'information en commun et de travailler en équipe tout en démontrant les caractéristiques mentionnées ci-dessus.

7. Comportement de poursuite - Un robot peut utiliser l'information d'an autre robot pour poursuivre ce dernier en temps réel.

En plus de ces résultats, des méthodes consistant à démontrer ces comportements autonomes en toute sécurité ont été créées et un niveau de base opérant de référenciation pour la mise au point robotique future a été établi. L'équipe de démonstration de robotique autonome a aussi été définie et consiste en un responsable de la sécurité durant les essais, un guide des essais, du personnel du soutien technique, des commissaires du terrain, un caméraman et un responsable de la collecte des données.

Portée des résultats et travaux futurs : Le logiciel Player/Stage ayant donné de bons résultats dans ce projet, la technologie des logiciels de source non secrète de commerce continueront de servir *de levier financier* pour RDDC, à l'avenir. De plus, un certain nombre de leçons ont été acquises concernant l'architecture de logiciels, telles que l'importance de la modularité et du concept de réseaux.

Au demeurant, des leçons ont été acquises concernant les difficultés de rendre les unités robotiques fonctionnelles à l'extérieur. Ces robots manquaient de stabilité et ont besoin de capteurs plus complexes, de mieux éviter les obstacles et de mieux planifier le parcours global pour être capables de fonctionner dans des scénarios plus réalistes. Il faut aussi s'efforcer de rendre les robots plus robustes et moins sensibles aux conditions environnementales.

Enfin, ces essais ont prouvé que les comportements coopératifs de base peuvent être atteints assez facilement en communiquant les positions par GPS au moyen d'un réseau de stations d'émissions radios. Ces capacités sont importantes mais le fait de dépendre du GPS et des communications explicites par radios pour la coopération est en quelque sorte indésirable car elles sont sujettes au brouillage intentionnel et au fonctionnement intermittent. À l'avenir, les comportements coopératifs devraient être basés sur la reconnaissance visuelle et l'établissement de la position relative, en utilisant aussi peu de largeur de bande que possible. Ceci augmentera la robustesse des équipes de véhicules terrestres sans pilotes.

Giesbrecht, J., Collier, J., Monckton, S. 2004. Staged Experiments in Mobile Vehicle Autonomy. DRDC Suffield TM 2004-288. R & D pour la défense Canada – Suffield.

Table of contents

List of figures

This page intentionally left blank.

1. Introduction

Through the Autonomous Land Systems (ALS) initiative Defence R&D Canada – Suffield is investigating the area of Unmanned Ground Vehicles (UGVs). The goal is to create autonomous robotic vehicles which can be useful in a wide variety of military applications without being a burden to their users. It is commonly assumed that providing vehicles to assist military personal need be a complex, expensive undertaking. However, this is not always the case. This project represents robotic abilities achievable with an inexpensive, rapid implementation by a small team, using off-the-shelf hardware and software. The project consisted of a set of simple tests designed not to provide a maximum amount of autonomy or robustness, but to demonstrate abilities achievable with a small platform, a small sensor suite and pre-existing software packages. The project successfully demonstrated useful autonomous behaviours and has shown that with a small amount more time and effort an even wider variety of relevant abilities could be developed.

This document gives an overview of the Segway RMP platform and all of the hardware and software components mounted on it, the specific trials conducted on Thursday, October 7th, 2004, and conclusions and lessons learned from the experiments.

1.1 Trial Objectives

The primary objective of the trial was to demonstrate vehicle autonomy achievable with off the shelf hardware and software and a small amount of development effort. A secondary goal of the trials was to familiarize DRDC Suffield staff with the technical problems of autonomous vehicle control and to demonstrate common approaches to robot navigation, with a goal of ramping the ALS team up in anticipation of a larger, far more complex demonstration in the fall of 2005.

A further goal of the project was to acclimatize staff with some of the safety issues associated with large robotic platforms, and with the process of demonstrating robotic autonomy. Thusly protocols focused on safe vehicle operations were necessary even for the smaller platforms used here. While avoiding human injury is the primary concern, damage to equipment is far more likely and can have serious consequences.

1.2 Trial Outline

With these goals in mind, the trials were composed of six phases that verified and characterized:

1. Vehicle Shutdown and Safety - Robot can be safely shut down and respond safely to a variety of hardware and software failures.
2. Teleoperation and Telemetry - Robot can be controlled remotely and can report back information such as laser scan readings, vehicle position, and robot state.

3. GPS Trajectory Following - Robot can be controlled in absolute position coordinates using GPS. Multiple way-points can be assigned to the robot for it to sequentially follow.

4. Dead-Reckoning - Robot can keep track of its position accurately.

5. Obstacle Avoidance - Robot can keep itself safe by avoiding both static and dynamic obstacles while making its way to goal locations.

6. Leader/Follower Behaviour - Robots are able to share information and work as a team while demonstrating all of the above characteristics.

7. Pursuit Behaviour - One robot can use information about another to pursue it in real-time.

1.3 Personnel

While it is technically possible to perform each experiment informally with one or two investigators, the technical value of field trials lies in the capture of performance data, which is often difficult to achieve with one or two staff. Therefore each experiment was conducted by an 'experiment team' and the experiment itself was composed of a series of pre-defined procedures detailing each step of the trial. The rationale behind this regimented structure is threefold: to ensure the safety of equipment and personnel, to maximize the value of the field trial through consistency, and to provide a foundation for future comparison. The team consists of the following:

1. *Experiment Safety Officer* - Jared Giesbrecht - The role of the ESO is to halt the experiment if any potential hazard to man or machine develops. Anyone at anytime may command a halt without repercussion, but an experiment will only commence with the permission of the ESO.

2. *Experiment Lead* - Jack Collier - The role of the EL is to manage the conduct of the experiment. Specifically, coordinating the execution of the experiment by calling out and verifying experiment check lists.

3. *Technical Support* personnel - Bruce Digney, Simon Monckton - The TS is charged with executing steps as called by the EL, such as turning on the vehicles, configuring software, etc.

4. *Area Marshals* - Ron Anderson - Marshals have a variety of roles from simply verifying in-bounds safety to setting up and removing obstacles, and collecting data.

5. *Camera Operator* - Mike Trentini - The camera operator documents the experiments with both video and still pictures.

6. *Data Collection Officer* - Isabelle Vincent - The Data Collection Officer is responsible for obtaining and organizing data, whether it be collecting physical measurements or labelling and organizing software log files.

1.4 Weather Monitoring and Restrictions

Because the platforms and hardware are not robust to weather conditions the trials needed to be conducted on a fair weather day. Forecasts were monitored on the Environment Canada website, and minimum conditions were as follows:

- No rain.
- Wind less than 30km/hr.
- Temperature above zero degrees Celsius.
- Dry ground.

1.5 Photographic and Video Records

Due to the subjective nature of some of the tests it was necessary to document the entire procedure with pictures and with video. This also served a second purpose in providing a visualization of the trial setup and of the tests conducted, if they should be repeated at a later date. It was the responsibility of the camera operator to ensure that sufficient documentation was captured.

1.6 Communications

The SpeedLan radios used in this experiment operate in the 2.3 to 2.45 GHz band, so no regulatory permission was required. In addition, it was ensured that there were no other users operating in the same frequency in the immediate test area on the test day, to remain free of radio interference.

2. Robot Platforms

The platforms used were two Segway RMPs (Robotic Mobility Platform) [2], pictured in Figure 1 below. They are two wheeled, self balancing, medium sized robots based upon the Segway HT (human transporter), which are useful for their large payload, long range and high speed capabilities (100lbs, 10 miles and 8 mph respectively).

2.1 Hardware

The sensors used for these tests comprised a relatively small suite, consisting of one SICK LMS200 laser scanner for obstacle detection, a Sokkia GMS2600 GPS unit for absolute position information, as well as the Segway platform's odometry sensors for dead-reckoning. Communication was accomplished through a TCP/IP radio mesh network via Wave Wireless SpeedLan radios. Processing power for the robot was a Laptop PC with a Pentium 4 2.0 GHz processor running Fedora Core 1 Linux, with 512 MB of RAM. It interfaced to the Segway via a PCMCIA to CANBus adapter card, to the SICK LMS200 laser via a USB/RS422 converter, to the GPS receiver via RS-232, and to the radio via built in Ethernet. Power was provided by 12 volt batteries regulated through a Vicor DC-DC converter which supplied both 24 and 12 volt power for the sensors.

To avoid over-complicating the project a host of other hardware was not included, specifically an IMU (Inertial Measurement Unit), an electronic compass, stereo cameras, colour video, etc. Nevertheless, with this small amount of hardware a great deal of autonomy was achieved and still more could be achievable given more development time.

Figure 1: *Segway Platform with Sensors .*

2.2 Software

Player/Stage[3] was used as a basis for controlling the robot and creating autonomy in this demonstration. Player is an open source server/client based software package which runs over TCP/IP networks. It was developed to support a variety of robot hardware and to provide a common interface for robot software development. Player also contains a great deal of support for networked multi-user, multi-robot interaction and control. Stage is a 2D simulation package which interfaces with the Player server.

The Player package consists of three main parts:

1. The server which runs on a processor directly connected to the robot and provides control for the robot, as well as a communications hub for all the devices and controllers in the system.

2. Drivers which are loaded at run time and provide the interface between specific devices (such as the Segway, the SICK laser and the GPS unit) and the Player server.

3. One or more Player clients which connect to the Player server either from the same processor or from a remote station to issue commands to the robot and retrieve data. This may be a user interface for teleoperation, or software which retrieves data or creates autonomous behaviour.

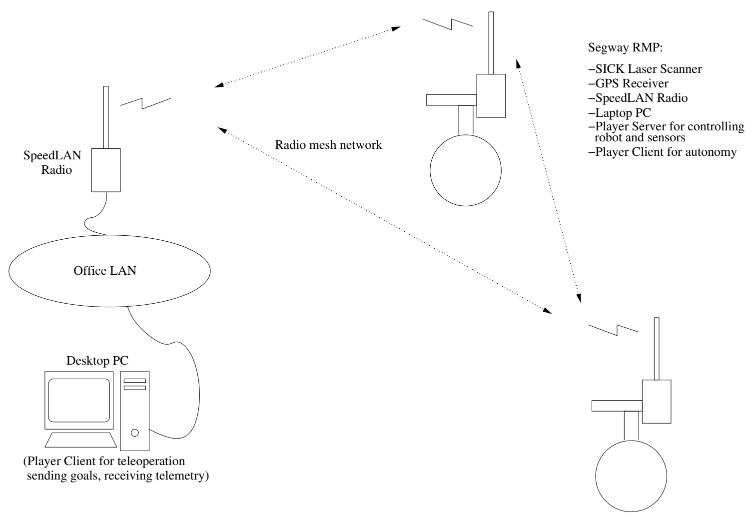

Figure 2: *System Overview.*

The Player project contains a number of pre-existing software modules such as teleoperation programs, obstacle avoidance algorithms, and drivers for controlling hardware like the Segway RMP, the SICK laser scanner, and GPS hardware. Given this base of software these demonstrations could be developed rapidly with only minor modifications to the Player drivers. The DRDC staff were then able to focus efforts on writing Player client software to create the autonomous behaviours desired.

With the system implemented, the Segways have six basic modes, as well as two advanced modes:

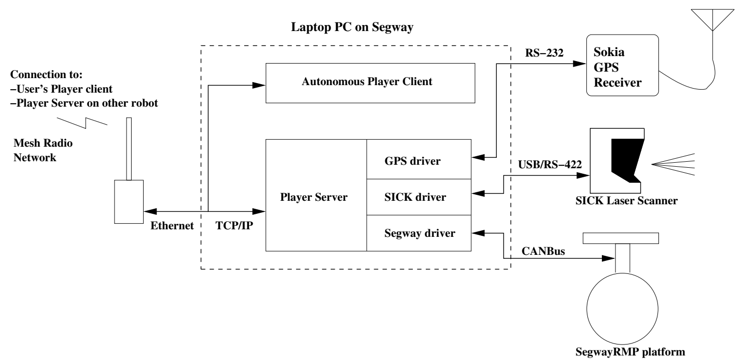

Figure 3: *Player Software on Robot.*

2.2.1 Basic Modes:

1. Balance Mode - The most basic state of the Segway. The robot is not controlled and does not move, but simply holds itself upright. All the other modes of robot operation begin from balance mode.

2. Teleoperation Mode - A Player server runs on the robot to provide direct control, and it receives commands from a Player client either on a remote machine, or on the same local host. All other control modes of the robot will be based upon this mode.

3. Obstacle Avoidance Mode - The robot is given an odometry goal location from a remote user and will travel to that position while avoiding obstacles.

4. GPS Mode - The robot is optionally given a GPS goal location from a remote user, and will travel to that position while avoiding obstacles. Otherwise, the GPS mode serves to keep the vehicles odometry correct with respect to the GPS readings.

5. Way-point Mode - The robot is given a series of coordinates to reach in a sequential order. This mode can be enabled with obstacle avoidance turned on or off.

6. GPS Way-point Mode - The robot is given a series GPS coordinates to reach in a sequential order. This mode can be enabled with obstacle avoidance turned on or off.

2.2.2 Advanced Modes:

1. Follower Mode -The robot will interrogate a leader robot to get the leader's current GPS location at a regular interval. These points will be used as a series of way-points which the follower must reach.

2. Pursuit Mode - The robot will interrogate a target robot to get the target robot's current GPS location at a regular interval and use this information to intercept it.

The following Player modules were used in this project:

1. One Player Server on each robot.
2. Separate Player Drivers for each of the following:
   (a) SICK Laser.
   (b) Sokkia GPS .
   (c) Segway RMP.
   (d) Vector Field Histogram Obstacle Avoidance.
3. A separate player client for each of the following:
   (a) Teleoperation and Laser Scan Visualization - PlayerView.
   (b) Goal setting and Odometry Correction by GPS.
   (c) GPS Way-point Navigation.
   (d) Leader Follower Behaviour.
   (e) Pursuit Behaviour.
   (f) Telemetry and Data Logging (Opstation QT).

3. Trial Procedure

The primary purpose of the Stage 1 trials was to benchmark 'State-of-the-Art' performance using existing open source robot control libraries (specifically Player Server 1.5) and an off-the-shelf vehicle (the Segway RMP). The secondary objective of the trial was to begin the evolution of experimental procedures towards consistent, repeatable, and informative autonomous vehicle experiments. In general the experiments were designed to verify key vehicle performance and, where applicable, characterize this performance for future reference. These experiments incrementally increased the degree of machine autonomy, from safety systems to autonomous way-point navigation, obstacle avoidance, and cooperative behaviour.

The trials described below were undertaken on the DRDC Suffield experimental proving ground, on a gravel parking lot adjacent to Building 34, as shown in Figure 4 on Thursday, Oct. 7th, 2004, and on a cement area inside Building 34 on Tuesday Nov. 23rd, 2004.

Figure 4: *Experimental Setup Outside Building 34.*

3.1 Safety Briefing

Before any work began a short overview of the operation of the Segway RMP was given. Most importantly, the Technical Support personnel were given instruction on how to operate the safety pull cord on the Segway. A viewing area for the spectators was created to the side of the demonstration, a safe distance away, and those in attendance were asked to remain in that area.

3.2 Vehicle Safety and Robustness

3.2.1 Purpose

The purpose of this experiment was to verify that platforms will stop under established conditions and to characterize the behaviour of the system during shutdown. Since subsequent trials had the vehicle moving over an unimproved gravel surface in close proximity to people and equipment, it was necessary to verify and characterize the shutdown of these systems in their possible running modes. In addition, it was desirable to characterize system performance and robustness in the face of various hardware failures such as processor or sensor failure.

3.2.2 Method

At the simplest level the systems must be powered up and then shut down. However, given the possibility of unknown shutdown behaviour and a variety of possible operational modes, the practical method for discovering these behaviours is, in fact, complex. The trial procedure is as follows:

1. *Area Ready* - When commanded by the EL, the experimental area must be cleared of all personnel with the exception of the experiment team. The area must be visually confirmed as clear. All team members should assume their roles. The ESO must confirm that the area is ready and inform the EL.

2. *Vehicle Enable* - The vehicle is enabled according to a published pre-start checklist. For the Segway this may include power decoupling, mode button depress, software initialization, etc.

3. *Vehicle Mode Select* - The vehicle must be placed into a known operating mode (again through a published checklist). There are a number of operational modes possible with the system, and this experiment will therefore simultaneously explore mode entry and exit.

4. *Vehicle Kill*

   (a) Balance Mode

      i. Lanyard Kill - The built in safety feature of the SegwayRMP is tested. It consists of a lanyard which powers down the vehicle when pulled to disconnect it from the platform.

   (b) Teleoperation Mode - PlayerView software (included with Player) is used to control the robot.

      i. Normal exit - The Player client is closed to end the control session.
      ii. Kill PlayerView - Simulation of a software crash of PlayerView.
      iii. Kill Player Server - Simulation of a software crash of the Player Server running on the robot host.
      iv. Loss of Signal on Wireless Connection - The connection between the base station and local host is broken (connection between Player client and server is lost).
      v. Loss of Signal on CANBUS - A disconnection of the communications between the SegwayRMP robot and the PC running the controlling Player server.

   (c) Player Obstacle Avoidance Mode

      i. Kill Player Client - The Player client on the base station which stimulated the autonomous behaviour is stopped.
      ii. Loss of SICK sensor - The SICK laser scanner which enables autonomous perception is shut down.

   (d) Player GPS Mode

i. Loss of GPS device - The GPS sensor fails entirely.

5. *Area Safe* - When the vehicle is disabled and verified by the ESO, the experimental area will be declared safe.

6. *Milestone Cleared* - If vehicle performance is deemed controllable by the team, the milestone will be considered cleared and recommended procedures will be identified for use in the next experiment. The team must be satisfied that the vehicle can be stopped safely under all conditions of the trial. If not, the limitations must be noted and accounted for.

3.2.3 Results

1. Balance Mode

   (a) Lanyard Kill - Pass - Vehicle powers down immediately when safety cord is pulled.

2. Teleoperation Mode

   (a) Normal Exit - Pass - Vehicle stops motion and remains stationary in balance mode.

   (b) Killed PlayerView - Pass - Vehicle stops motion and remains stationary in balance mode.

   (c) Killed Player Server- Pass - Vehicle stops motion and remains stationary, in balance mode.

   (d) Loss of Signal on Wireless Connection - Pass - Vehicle stops motion and remains stationary, in balance mode.

   (e) Loss of Signal on CANBus - Pass - Vehicle stops motion and remains stationary, in balance mode.

3. Player Obstacle Avoidance Mode

   (a) Killed Player Client - Pass - Vehicle stops motion and remains stationary, in balance mode

   (b) Loss of SICK Sensor - Fail - With the loss of the laser scanner the system retains its last set of sensor data indefinitely, and will act accordingly. For example, if the last laser scan indicates an object to the vehicle's left, it will continue turning right forever, reacting to the "ghost" obstacle from the last set of laser data.

3.2.4 Discussion

Using the Player server as a platform for control had a definite benefit for robustness in the face of the failure modes. Anytime a controlling client is disconnected from the server, the server immediately tells the robot platform to stop. In addition, the vehicle also stops reliably on the crash of the controlling server, or of any failure in the system communications. This can be undesirable behaviour in the face of intermittent communications, but for safety purposes in this system it served well.

One big problem with the Player system was its behaviour with the loss of the SICK sensor. The behaviour generated was very poor, and unsafe. However, with a small amount of effort, the system could be modified to check the time-stamp of the data generated, and set to halt operation if the data is too old.

During the course of the experiment is was also discovered that if the Segway was restarted while there were processes running which hadn't been killed, the Segway would respond to their commands. This was an unsafe behaviour of this system which must be dealt with. Testing of this behaviour will be incorporated into future trials.

3.3 Teleoperation and Telemetry

3.3.1 Purpose

The purpose of this experiment was to verify that platforms receive basic teleoperation commands and return basic telemetry data. This is the base technology necessary for robotic platforms. In this trial the teleoperation capabilities consisted of controlling robot speed and direction while displaying and logging data from the robot's odometry as well as the other on-board sensors.

3.3.2 Method

In summary, the system must be powered up, enabled, set into a teleoperation mode, set to specific maneuvers, and then shut down. Teleoperation is through keyboard and mouse via the PlayerView software included with Player. Telemetry data is gathered and logged via the Opstation QT program written at DRDC Suffield, which connects to the Player server on the robot. The trial procedure is as follows:

1. *Area Ready.*
2. *Vehicle Enable.*
3. *Vehicle Teleoperation Mode Select*.

4. *Vehicle Teleoperation Exercise(s)* - The intention is to incrementally explore the vehicles performance envelope as a guideline for future vehicle maneuver restrictions.
   (a) Explore safe range of motion - forward/reverse/CW/CCW.
   (b) Incremental increase in velocity using the following maneuvers, watching for unstable rocking modes (roll/pitch/yaw).
      i. Forward/reverse.
      ii. CW/CCW stationary rotation.
5. *Vehicle Telemetry Exercise(s)* - While undertaking the activities from the previous step, the user must be able to view and log GPS position, odometry data, laser scans, and robot state (battery power and velocity).
6. *Vehicle Kill.*
7. *Area Safe.*
8. *Milestone Cleared* - The team must be satisfied that the Player Client/Server connection is safe to use.

3.3.3 Results

1. Teleoperation Exercise
   (a) Safe Range of Motion - Pass - All movements could be controlled reliably.
   (b) Incremental Increase in Velocity - Pass - User could control the velocity of the Segway reliably in all directions.
2. Telemetry Exercise
   (a) View data - Pass - Data for all of the items listed was updated quickly and reliably.
   (b) Log data - Pass - Data for all of the items listed was logged reliably with time-stamp.

3.3.4 Discussion

The PlayerView software (see Figure 5) worked well to control the vehicle. All of the testing was done with the robot speed limited to 1 meter/second. Within this limited range, we were able to accurately control the dynamic performance of the vehicle in teleoperation mode. In addition, the Opstation QT software worked well for both visually displaying and logging the data.

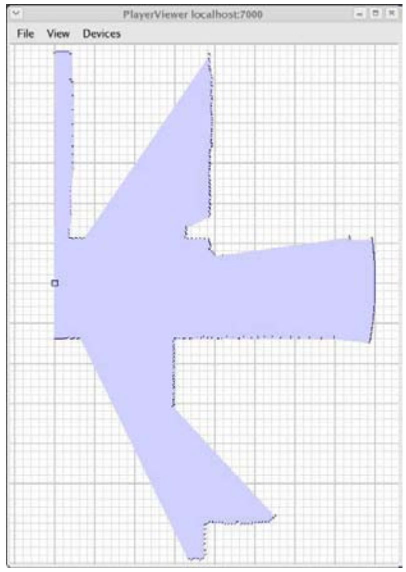

Figure 5: Laser scan information in PlayerView.

3.4 Deduced Reckoning

3.4.1 Purpose

The purpose of this experiment is to reveal systematic and non-systematic odometry errors. The procedure is based upon a standard dead-reckoning test used in the robotics community called the UMBmark test[1]. GPS sensing was not be used, and unlike the GPS test results there can be statistical value in the results of the dead-reckoning test. It would be feasible to extract both the systematic errors present in the SegwayRMP platform as well as non-systematic errors introduced by the environment. With this in mind, further navigation system development using additional sensors (IMU, gyroscope, etc.) can be bench-marked against this first test.

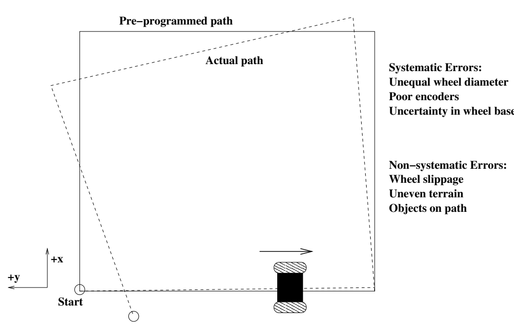

Figure 6: Dead-Reckoning Test

3.4.2 Method

In summary, the system must be powered up, enabled, given a set of way-points, set into a way-point following mode, and then shut down. The way-points will be a set of specific locations in the area forming a square 4mx4m in dimension, as shown in Figure 6. The test was conducted indoors on a smooth concrete floor. GPS will be disabled for this trial. The trial procedure is as follows:

1. *Area Ready.*

2. *Vehicle Enable.*

3. *Vehicle Way-point Mode Select.*

4. *Vehicle Deduced Reckoning Way-point Exercise* -This experiment consists of commanding the robot on a square route (4m x 4m) over two runs, one clockwise and one counterclockwise. This process is repeated 5 times. The absolute position of the robot is recorded at the end of each clockwise or counterclockwise run.

5. *Vehicle Kill.*

6. *Area Safe.*

7. *Milestone Cleared* - As before, if vehicle deduced reckoning is deemed satisfactory by the team, the milestone is considered cleared and recommended procedures will be identified for use in the next experiment.

3.4.3 Results

The data accumulated for this test is shown in Figure 7. The intended goal of the robot is at the (0,0) coordinate in the graph, and the actual positions where the robot calculated it had reached the goal are displayed.

3.4.4 Discussion

As can be seen from the two runs there is a large systematic and a small non systematic error in these results. This is indicated by the tight clustering of the data points for the clockwise and counterclockwise traversals. Given that the end locations for the two different directions are so widely separated, it can also be concluded that the error is due to odometry inaccuracy while turning. This conclusion is supported by common experience in the robotics field in which heading error from vehicles making tight turns is a great deal more significant than the translational error accumulated over the run. This would explain the results seen here.

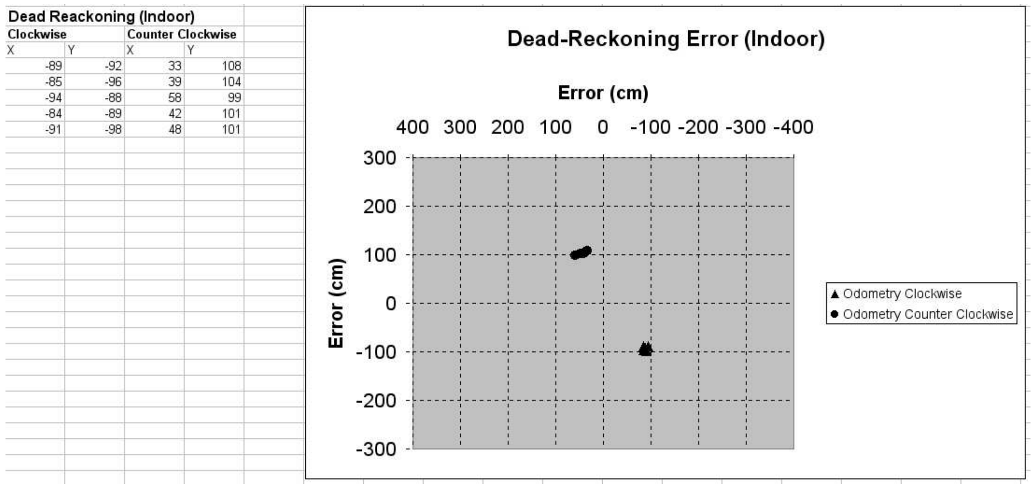

Figure 7: *Dead-Reckoning Results (4m x 4m smooth concrete).*

3.5 GPS Trajectory Following

3.5.1 Purpose

The purpose of this experiment was to verify that platforms can successfully follow a set of GPS way-points and to characterize the manner in which this occurs. The absolute position of the robot in the test was measured and recorded to ensure that the robot performs as desired, but the results will be of little statistical significance given the randomly variable nature of GPS signals due to atmospheric conditions, clock drifts and RF multipath effects. A second trial was undertaken with no GPS correction, using only odometry. The results of this test were included as a reference as an improvement offered by the GPS.

3.5.2 Method

In summary, the system must be powered up, enabled, given a set of way-points, set into a way-point following mode, and then shut down. The way-points are a set of specific locations in the area. This trial was conducted outdoors on a gravel parking lot. The trial procedure is as follows:

1. *Area Ready.*
2. *Vehicle Enable.*
3. *Vehicle GPS Way-point Mode Select.*
4. *Vehicle Way-point Exercise* - This experiment consists of commanding the robot on a square route (10m x10m) in both a clockwise and counterclockwise direction. This process is repeated 5 times. The

absolute position of the robot is recorded at the end of each clockwise or counterclockwise run. This whole process is then repeated with the GPS correction disabled to serve as a reference.

5. *Vehicle Kill.*

6. *Area Safe.*

7. *Milestone Cleared* - The team must be satisfied that the vehicle can move from one GPS way-point to another, with a reasonable degree of accuracy.

3.5.3 Results

The data accumulated for this test is shown in Figure 8. The intended goal of the robot is at the (0,0) coordinate in the graph, and the actual positions where the robot calculated it had reached the goal are displayed.

GPS Waypoint			
Clockwise		Counter Clockwise	
X	Y	X	Y
-4	102	-73	6
-123	89	-103	23
-34	-126	-153	41
-32	-114	-151	21
-158	-117	-116	65

Dead Reackoning			
Clockwise		Counter Clockwise	
X	Y	X	Y
169	-52	-152	282
217	-55	-143	320
260	-54	-145	337
206	-78	-122	315
220	94	-112	321

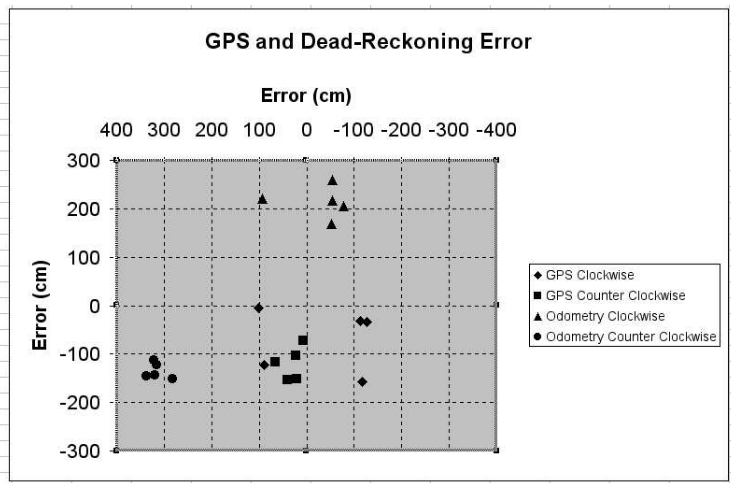

Figure 8: *GPS and Dead Reckoning Error (10m x 10m gravel square).*

3.5.4 Discussion

Similar to the dead-reckoning results indicated earlier, the dead-reckoning (odometry) results for this test had a large systematic and small non-systematic error, as the data points are tightly clustered around an inaccurate location. This is, again, most likely to to a lack of angular heading resolution while turning. Note that the non-systematic error was greater than for the indoor trials in the previous section, as would be expected on a gravel parking lot. However, with GPS correction the large systematic error was almost entirely removed. In its place, we find an increased non-systematic error (i.e. the data is more accurate, but less precise). This is what we would expect from the GPS system, which will give us absolute position correction, but is subject to more random drift.

Because our GPS system did not provide a heading correction, only position correction, the robot would wander off course in between the way-points as badly as it did using only odometry. However, the GPS places an absolute bound on the error, and the Segway would eventually return to within the stated tolerance at each of the way-points. This highlights the need for an absolute heading sensor to be included along with an absolute x,y sensor on any robotic navigation system, using an electronic compass or an INS. Although the GPS itself is capable of providing heading, it can only do so when the GPS is in motion and therefore initial heading error is inevitable. Another option would be to use sequential GPS points to provide heading. This is not fool proof, as it will give erroneous results when the vehicle is turning, and also depends on the accuracy of the GPS sensor. With the use of a Differential GPS system (DGPS), heading from sequential positions will generally be a viable solution due to the greatly increased accuracy of that system. In order to overcome the turning problem, we could program the system to only do absolute heading corrections when the vehicle is travelling in a straight line.

3.6 Obstacle Avoidance

3.6.1 Purpose

The purpose of this experiment was to verify and characterize the obstacle avoidance capabilities of the platform while pursing the goal location. The software used in this trial was based upon the Vector Field Histogram algorithm[4]. The algorithm retrieves data from a 180 degree horizontal scan of the environment from the SICK laser scanner, and then uses this data to proceed in the most effective direction, trading off safety with a desire to reach the goal. The algorithm runs many times per second, whenever new laser data is available, to provide fast reaction to both static and dynamic obstacles. There are a number of parameters which can be modified in the algorithm to provide either more cautious or aggressive actions by the robot. They were adjusted ahead of this trial to provide safe yet goal directed behaviour at moderate robot speeds. This trial verifies that the robot is indeed safe to operate in a world populated by both solid obstacles as well as moving people.

3.6.2 Method

In summary, the system must be powered up, enabled, given a target objective in GPS coordinates and set into an obstacle avoidance mode. The procedure is as follows:

1. *Area Ready* - When commanded by the EL, the experimental area must be

cleared of all personnel with the exception of the experiment team.

2. *Vehicle Enable.*

3. *Vehicle GPS Mode Select.*

4. *Vehicle Obstacle Avoidance Exercise.*

   (a) Static obstacle field - The robot is given a goal location on the opposite side of the obstacle field, 20m from the start position, with the intention that the easiest, most goal directed route will be through the obstacles. The field consists of 10 static obstacles, arranged as shown in Figure 9.

   (b) Dynamic obstacle field - The goal is once more given 20m straight ahead of the robot. A human walking at normal speed will be the dynamic obstacle in this trial.

      i. Dynamic test 1 has the human start at the goal and walk directly towards the robot in its path to the goal (Figure 10).

      ii. Dynamic test 2 has the human walk on a path which intersects the robot at a distance in front of it, but which should not collide. This is repeated from both directions, with intersections at about 1 meter and 2 meters in front of the robot.

      iii. Dynamic test 3 has the human walking on a collision course with the robot. It will be repeated twice, once from either side.

5. *Vehicle Kill.*

6. *Area Safe.*

7. *Milestone Cleared* - The milestone is cleared when the robot has reached the goal location without coming into contact with either static obstacles, or humans moving in the obstacle field. No specified time limit will be given to reach the goal successfully in any of the trials.

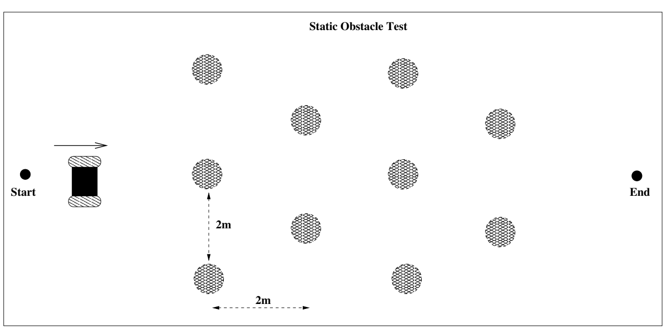

Figure 9: *Static Obstacle Avoidance.*

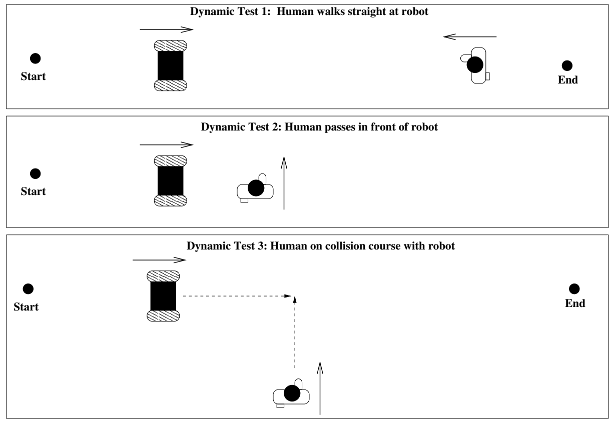

Figure 10: *Dynamic Obstacle Avoidance.*

3.6.3 Results

1. Static Obstacle Field - Pass - The obstacle avoidance performed very well in the obstacle field. No obstacles were contacted and the goal was reached rapidly and directly.
2. Dynamic Test 1 - Pass - The Segway successfully avoided the head on collision by dodging around the person, and continuing on its way smoothly.
3. Dynamic Test 2 - Pass - The Segway successfully avoided the human. The robot turned in the same direction as the human was travelling , and would move parallel to the person, until the person moved past, and the robot could continue towards the goal.
4. Dynamic Test 3 - Pass - The Segway successfully avoided the collision, and behaved in a similar way as in the previous test.

3.6.4 Discussion

The Vector Field Histogram algorithm, which performed obstacle avoidance for the Segway, has a number of parameters which can be tuned to enhance its behaviour. A user can make the system more aggressive or more cautious. This means that you can always get it to work well, but in a limited way. For example, if you want smooth behaviour, the system is tuned one way, but if you want aggressive avoidance of obstacles, you must forgo the smooth behaviour when obstacles are detected. A great deal of this is due to the limited look ahead of the SICK laser scanner, but the effect is also produced because the algorithm is not a deliberative, look-ahead path planner. It simply reacts to the sensor data at each scan. With a small amount of experimentation, we were able to create a system with a favorable balance between smooth and reactive behaviour. However, we were only operating the robot at low robot speeds (maximum of 1 m/s). The Vector Field Histogram has no consideration for the dynamic stability of the platform at higher speeds. It may have been more difficult to create a stable algorithm which could control the vehicle smoothly, while still avoiding obstacles at high speeds without tipping the Segway over. This is a limitation of the VFH system, which is quite primitive in comparison to some dynamic path planners which account for vehicle kinematics and dynamics.

A second limitation of VFH is that it is not intended for dynamic obstacles. It has no provision for estimating the motion of the obstacles it has detected or predicting their future position. All obstacles are treated as static. In order to avoid dynamic obstacles it must rely on the fact that it begins to avoid an obstacle when it first sees it farther out, and then will steer away from it with more urgency as the object becomes closer. Despite this fact it seemed to perform well in avoiding the moving person in the above experiments as long as the person was travelling at a normal walking pace. It was not as intelligent as a human controller may have been, but it didn't have any problem avoiding the dynamic obstacles.

There are also some other limitations to the obstacle avoidance system as we have implemented here, such as its limitation to two dimension obstacle detection and avoidance. The SICK laser scanner was mounted to aim completely horizontally and no nodding mechanism was used. All of the obstacles we used and tested were tall enough to be picked up by the horizontal scan of the SICK laser on top of the Segway and did not protrude at all below this level. However, this is not a very realistic consideration for a vehicle such as this which is relatively tall and unstable.

If we set aside the limitations given above the VFH system was very robust and performed well. If it were to be used on a lower, more stable platform, for indoor applications, it would work well. However, for outdoor unmanned ground vehicles a much more complex system will need to be developed.

3.7 Leader/Follower Behaviour

3.7.1 Purpose

The purpose of this experiment was to show an autonomous behaviour achievable with the simple hardware and software implemented. The robots shared GPS location and it was used to allow a second robot to follow the path created by the first. In order to do this, a second robot was added to the trials, which used the same basic system as demonstrated so far (Autonomous Goal Mode, Way-point Mode, etc). Two tests were undertaken for the leader/follower behaviour. The first used a simple route taken around obstacles by the leader robot to reach a goal. The second test used a human teleoperator on the lead vehicle, taking a route of his/her choosing through the obstacle field.

3.7.2 Method - Simple Route

The procedure is as follows:

1. *Area Ready* - When commanded by the EL, the experimental area must be cleared of all personnel with the exception of the experiment team.
2. *Vehicle Enable.*
3. *Vehicle Mode Select* - The leader vehicle and follower vehicles are placed in GPS Mode.
4. *Leader Goal Instruction*- The lead robot is given a goal location 30m straight ahead of it, with part of the straight line path obscured by obstacles.
5. *Follower Behaviour Initiated* -The follower starts at a different location from the leader and is given no goal at all by the human user. Once the leader has begun its route, the Follower mode is started on the second robot. It interrogates the leader for its position every 5 seconds and uses those positions as sequential way-points which it then follows. In this manner the follower will follow the approximate route that the leader took to reach the goal.
6. *Vehicle Kill.*
7. *Area Safe.*
8. *Milestone Cleared* - The milestone is cleared when both the leader and follower robots have reached the goal location. In addition, the follower robot must have followed the leader's path within the reasonable bounds given by the GPS system accuracy.

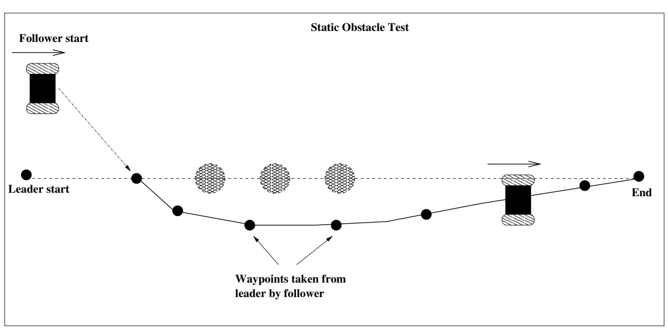

Figure 11: *Leader/Follower Behaviour.*

3.7.3 Method - Complex Route with Obstacles

The procedure is as follows:

1. *Area Ready* - When commanded by the EL, the experimental area must be cleared of all personnel with the exception of the experiment team.

2. *Vehicle Enable.*

3. *Vehicle Mode Select* - The leader vehicle is placed in Teleoperation Mode, and the Follower is placed in GPS Mode.

4. *Leader Goal Instruction-* The lead robot is driven through the obstacle field on a route of the human's choosing.

5. *Follower Behaviour Initiated* -The follower starts at a different location from the leader, and is given no goal at all by the human user. Once the leader has begun its route, the Follower mode is started on the second robot. It interrogates the leader for its position every 5 seconds, and uses those positions as sequential way-points, which it then follows. In this manner, the follower will follow the approximate route that the leader took to reach the goal.

6. *Vehicle Kill.*

7. *Area Safe.*

8. *Milestone Cleared* - The milestone is cleared when both the leader and follower robots have reached the goal location. In addition, the follower robot must have followed the leader's path within the reasonable bounds given by the GPS system accuracy.

3.7.4 Results

1. Simple Route - Pass - The second robot could follow quite closely the path taken by the first.

2. Complex Route with Obstacles - Pass - The second robot could follow the path taken by the first. However the robot did deviate to a small degree to avoid obstacles in a slightly different way than the lead robot had. However, the general route was the same, and the follower robot was able to reach the goal position which was specified only by the lead robot, with no human intervention.

3.7.5 Discussion

With Player server and GPS technologies this behaviour was very easy to create. It also seemed to be fairly robust and the system had no problem avoiding obstacles while being goal directed. However it would be much better to be able to create a cooperative behaviour not reliant on GPS and radio communications, instead using visual recognition of the target. This would mean less reliance on GPS which is jammable and unavailable indoors. It would also mean less bandwidth usage for the system, which can become important when large numbers of vehicles are used in an operation.

3.8 Pursuit Behaviour

3.8.1 Purpose

The purpose of this test was very similar to the Leader/Follower behaviour experiment. However, in this test, rather than follow the route which the leader has chosen, the second robot pursues the first robot's position directly. This behaviour was demonstrated in two ways. Firstly, with the first robot travelling along an open square route, by way-point navigation, and secondly with the first robot travelling an amorphous route through the obstacle field described earlier, by user teleoperation.

3.8.2 Method - Simple Route

The procedure is as follows:

1. *Area Ready* - When commanded by the EL, the experimental area must be cleared of all personnel with the exception of the experiment team.
2. *Vehicle Enable.*
3. *Vehicle Mode Select* -Both vehicles are placed in Obstacle Avoidance mode.
4. *"Mouse" Goal Instruction* - A "Mouse" robot is set on a square route 20m x 20m.
5. *"Cat" Behaviour Initiated* - After a delay of 10 seconds, the "Cat" is set on a pursuit by starting the Pursuit mode. The Cat interrogates the Mouse for its position once per second.
6. *Vehicle Kill.*
7. *Area Safe.*
8. *Milestone Cleared* - The milestone is cleared when the "Cat" has gotten within a 1 meter radius of the "Mouse", and it declares that it has "caught the mouse". Both robot's then stop.

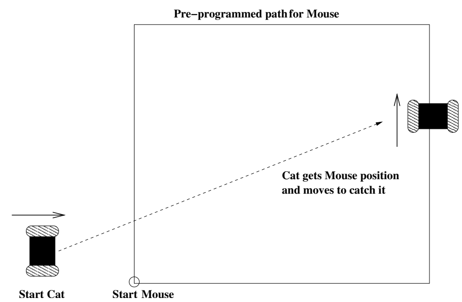

Figure 12: *Simple Pursuit Behaviour*

3.8.3 Method - Complex Route with Obstacles

The procedure is as follows:

1. *Area Ready* - When commanded by the EL, the experimental area must be cleared of all personnel with the exception of the experiment team.
2. *Vehicle Enable.*
3. *Vehicle Mode Select* -Both the vehicles are placed in Obstacle Avoidance mode.
4. *"Mouse" Goal Instruction* - The "Mouse" is given a series of way-points within the obstacle field, which the second robot follows, while avoiding obstacles.
5. *"Cat" Behaviour Initiated* - After a delay of 10 seconds, the "Cat" is set on a pursuit by starting the Pursuit mode. The Cat interrogates the Mouse for its position once per second.
6. *Vehicle Kill.*
7. *Area Safe.*
8. *Milestone Cleared* - The milestone is cleared when the Cat has gotten within a 1 meter radius of the Mouse, and it declares that it has "caught the mouse". Both robot's then stop.

3.8.4 Results

1. Simple Route - Pass - The "Cat" easily took the shortcut and intercepted the "Mouse"
2. Complex Route with Obstacles - Pass - The "Cat" eventually caught the "Mouse", but had some difficulty due to its cautious evasion of the intervening obstacles.

3.8.5 Discussion

This behaviour was also fairly easy to implement with the Player Server and GPS. Due to the inaccuracy of the GPS, the “Cat” tended to suffer from sub-goal obsession. It would succeed in getting close to the robot, stop, and then restart the pursuit again if the GPS drifted. As with the leader/follower test, a much more desirable behaviour would be to utilize visual target recognition to create this behaviour.

4. Conclusions and Lessons Learned

In undertaking these experiments we were able to attain a fair amount of autonomy and cooperative behaviour using off the shelf hardware and software, and a small amount of development time. However, these systems were nowhere near to being field ready units. Rather than relying solely on 3rd party solutions it will be beneficial for DRDC Suffield to leverage the work of others in our robot development. Experiments such as these allow us to learn many lessons about systems software, platform effectiveness, sensing, perception, localization, and cooperative behaviours, which will assist in our development of UGVs in the near future. Several of these lessons are further discussed below.

4.1 Systems Software

The Player/Stage open source software project was a very useful tool for implementing this robotic system. In undertaking these experiments, we were able to attain a fair amount of autonomy and cooperative behaviour with only a small amount of development time. The modularity and well thought out design allowed the developers to easily add new capabilities to the existing system and load them at run time. Furthermore, the network centered design of Player allowed us to move processes to where processing power exists. In addition, communication between different software agents was easily achievable through Player’s network centered design.

Although the Player system generally performed well for our purposes, there were a couple of safety issues which were revealed by the experiments. Firstly, we found that Player did not terminate its processes when the vehicle was shut down. This would cause the vehicle to experience undesirable behaviour when it was turned on again. This was because Player server was still generating commands which the Segway responded to when it was turned back on. Any UGVs developed by ALS should ensure that all processes are terminated when the vehicle is shut down, by providing software watchdogs to manage system processes. Secondly, the Player server and drivers which we used were extremely sensitive to error states. A great deal of work would need to be done to make the systems robust, such as accurate error detection and reporting, as well as software recovery mechanisms to maintain autonomy. These issues must be dealt with for any UGVs developed by ALS.

Another disadvantage of the Player system is its lack of network robustness. Although for the purposes or our experiments the networking capability was generally sufficient, we did encounter some interference and drop out problems. More practical robot systems must be able to overcome these network problems. They would need to act appropriately in the face of network dropout and be able to cope with a continually changing network topology. Player's reliance on a priori knowledge of all the IP addresses in the system is also a serious downfall which must be dealt with. This requirement means that a system cannot be changed dynamically without recompiling software. Tools such as CORBA have capabilities which will allow greater network robustness.

4.2 Platform Effectiveness

The Segway platform was chosen for this experiment as there was already a large body of software written to work with the platform. While carrying out these experiments we were able to make a number of observations about the Segway platform. While these observations are platform specific they do highlight a number of important areas of concern for any UGV. The most observable characteristic of the Segway platform was its instability on rough ground. This was especially apparent in the side to side direction. As a result we chose to run the Segway at very low speeds to avoid tipping. While it is reasonable to assume that a four wheeled ATV sized platform would be inherently more stable, it is important to note that these experiments were undertaken on a relatively smooth gravel parking lot. A more complex obstacle avoidance and navigation system will be needed to control the vehicle in a stable manner over rough terrain. In addition, we would need to add a global path planner with higher intelligence to prevent the robot from being trapped in cul-de-sac situations.

Another important consideration is susceptibility to environmental conditions. The Segway platform and its associated hardware was not very weatherproof and therefore at the mercy of the elements. Any military relevant UGVs need to be ruggedized to allow all weather operational capability.

4.3 Sensing, Perception, and Localization

Although the use of one SICK laser scanner for all environmental sensing was sufficient for these experiments, it also limited the capabilities and effectiveness of the behaviours we were able to design. In particular, the system was vulnerable to obstacles which were just beyond the single sensor's field of view. As the robot turned the obstacle would suddenly appear and the Segway would have to react quickly to avoid the obstacle. While the platform was able to deal with these obstacles, system performance could be greatly increased by added multiple sensors at different view points to increase the robot's awareness of its surroundings. Furthermore, in the complex battlefield environment a wider suite of sensors will be needed to accommodate different types of terrain, vegetation such as grass, bushes, and visibility conditions such as smoke, fog and low light. Sensors such as sonar, stereo vision, and ultrasonic detection are possible additions. Finally, the 2D navigation system used for these experiments was found to be very limiting and is not acceptable for a real UGV system in an outdoor environment. We need to sense the world in both the horizontal and vertical directions, and add in metrics of traversability and roughness.

The robot localization experiments revealed the inaccuracy of odometry systems, due to their susceptibility to drift. In addition, wheel slippage caused further error in the odometry system. In particular the Segway was susceptible to heading errors when turning. To help combat these problems GPS was used to supplement the data. Future localization techniques will need to be developed further using better fusion of data from different sensor sources, correction of error through the use of landmarks, or the addition of an absolute heading reference.

4.4 Cooperative Behaviours

These experiments proved that basic cooperative behaviours can be achieved fairly easily through the communication of GPS positions through a radio network. Although these are important capabilities, reliance on GPS and explicit radio communications for cooperation is somewhat undesirable as they are subject to jamming and intermittent availability. In the future, cooperative behaviours should be based on visual recognition and relative position finding, with the use of as little bandwidth as possible. This will increase the robustness of teams of UGVs.

4.5 Future Considerations

We were able to achieve quite a good level of autonomy using off the shelf hardware and software, with only a little time and effort to develop. Ultimately this validates the idea of leveraging existing UGV solutions to enhance our capability. Off the shelf software technology will continue to be be leveraged to create even more advanced behaviours in the future. However, we have shown that existing solutions do have limitations which must be dealt with. Examining these limitations will allow us to develop our own UGVs with these issues in mind. The primary lesson we can take from the preceding discussion is the need for system robustness, whether it be communications, error detection, or coping with environmental conditions. This concept along with the new wealth of knowledge we have gained from these experiments will be directly applied to future UGVs developed at DRDC Suffield.

5. Appendix A: Implementation Details

5.1 Hardware:

1. PC: Laptop PC, Pentium 4 2.0 GHz processor with Fedora Core 1 Linux, Kernel 2.4.22-1.2197 and 512 MB of RAM

2. Robot Platform: Segway RMP interfaced to PC with Kvaser Lapcan PCMCIA/CANBus card

3. GPS: Sokkia GMS2600 with antenna, interfaced to PC using RS232 on PC serial port

4. Laser Scanner: SICK LMS200 interfaced via a National Robotics Engineering Consortium RS232/RS422 Converter board, and a EasySync US232 RS232/USB converter to USB port on the PC

5. Communications: SpeedLan 9000 radio, interfaced to PC Ethernet and 24 volt supply with Wave Wireless junction box (PN 1000-3754

6. Power: 12Volt, 8 AHr Lead-Acid battery,with a Vicor 12-24Volt DC-DC Converter

7. Power: Internal batteries in Laptop PC and Segway RMP

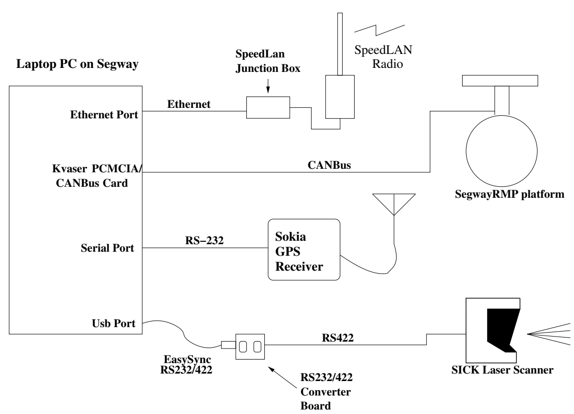

Figure 13: *Sensor Wiring*

5.2 Software:

1. Player Server Version 1.5

2. Player drivers included with Release 1.5 for SICK LMS200, GPS, Segway RMP, and Vector Field Histogram Obstacle Avoidance

3. Player client included wit Release 1.5 - PlayerView for Teleoperation and Telemetry

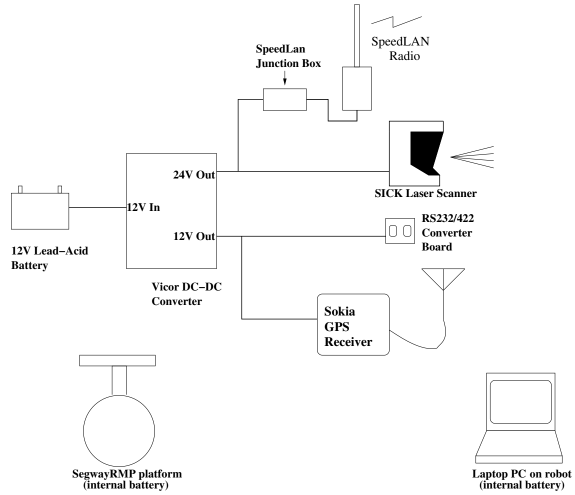

Figure 14: *Power Wiring*

4. Player clients written by DRDC (files available on DRDC Suffield TVSS Concurrent Versioning Systems (CVS) repository):

   (a) Goal setting and odometry correction by GPS: vfh_jared.cc

   (b) GPS Waypoint Navigation: multi_waypoint_gps.cc

   (c) Leader Follower Behaviour: leader_follower.cc

   (d) Pursuit Behaviour: cat_and_mouse_client_segway.cc

   (e) Telemetry and data logging - OpstationQT

6. Appendix B: Trial Performance Checklist

6.1 Installation and Setup

Installing player, files to mod, etc

Setup of radios:

Radio #1 gets 131.135.76.193

PC on Segway #1 gets 131.135.76.195

Radio #2 is 131.135.76.177

PC on Segway #2 is 131.135.76.178

Running opstation_qt procedure:

1. type in IP of robot
2. create client
3. connect to player
4. create devices
5. connect to devices
6. click on log data check box for each appropriate item

6.2 Vehicle Safety and Robustness

6.2.1 Balance Mode

1. Put vehicle in balance mode (Hold Segway upright, put key in Segway starter, press blue Segway button once)
2. Pull lanyard

6.2.2 Player Teleop Mode

1. Put Segway in balance mode
2. On robot PC(player/config): $player Segwayrmp.cfg
3. On base PC: run $playerv -h 131.135.76.195
4. On base PC: In playerv: Devices->Position (0)->Subscribe, Devices->Position (0)->Command, Devices->Position (0)->Enable

5. On base PC: In playerv: Use interface to verify motion control of robot
6. On base PC: In playerv: File->Exit ... Ensure robot motion stops
7. Repeat steps 1-5. On base PC: In playerv: Use CNTRL-C to kill program (Client crash)... Ensure robot motion stops
8. Repeat steps 1-5. On robot PC: In player: Use CNTRL-C to kill program (Server crash)... Ensure robot motion stops
9. Repeat steps 1-5. Remove power to robot PC ... Ensure robot motion stops
10. Repeat steps 1-5. Remove power to base PC ... Ensure robot motion stops
11. Repeat steps 1-5. Remove power from radio ... Ensure robot motion stops
12. Repeat steps 1-5. Disconnect CANBUS cables from robot ... Ensure robot motion stops

6.2.3 Player Obstacle Avoidance Mode

1. Put Segway in balance mode , start SICK scanner, GPS unit, radio
2. On robot PC (player/config):$player Segwayrmp_vfh.cfg
3. On base PC (player/examples/c++): run $./vfh -h 131.135.76.195 -m -g 20 0 0
4. On base PC: In vfh: Use CNTRL-C to kill program (Client crash)... Ensure robot motion stops
5. Repeat steps 1-3. Remove power to base PC ... Ensure robot motion stops
6. Repeat steps 1-3. Remove power from radio ... Ensure robot motion stops
7. Repeat steps 1-3. Remove power from SICK laser ... Ensure robot motion stops

6.2.4 Player GPS Mode

1. Put Segway in balance mode
2. On robot PC (player/config): $player segwayrmp_vfh.cfg
3. On base PC: run $./vfh_jared -m -h 131.135.76.195 -g <appropriate northings> <appropriate eastings>
4. Remove antenna from GPS unit so it loses GPS fix. Ensure robot stops and waits to reacquire signal. Robot motion should begin again when GPS is reattached and fix reacquired.

5. Repeat steps 1-3. Remove power from GPS unit ... Ensure robot motion stops

6.2.5 Player Waypoint Mode

6.3 Teleoperation and Telemetry

1. Put Segway in balance mode, start SICK scanner, start GPS unit
2. On robot PC(player/config): $player -d telemeter.so segwayrmp.cfg
3. On base PC: run $playerv -h 131.135.76.195
4. On base PC: In playerv: Devices->Position (0)->Subscribe, Devices->Position (0)->Command, Devices->Position (0)->Enable, Devices->Laser (0)->Subscribe
5. On base PC: In playerv: Use interface to verify motion control of robot
6. On base PC: In playerv: Verify that laser scan data is being displayed accurately.
7. On base PC: run $./opstation_qt ... Current odometry and gps data should display to the screen, as well as log to file.

6.4 Deduced Reckoning

1. Put Segway in balance mode, start SICK scanner, start GPS unit
2. On robot PC(player/config): $player -d telemeter.so segwayrmp.cfg
3. On robot PC: $./vfh -g 0 0 0
4. On robot PC: run $multi_waypoint. Add odometry goals as required ... Verify robot moves to goal locations as desired. Record accuracy.
5. On base PC: run $./opstation_qt ... Current odometry and gps data should display to the screen, as well as log to file.

6.5 GPS Trajectory Following

1. Put Segway in balance mode, start SICK scanner, start GPS unit
2. On robot PC(player/config): $player -d telemeter.so segwayrmp.cfg
3. On robot PC: run $vfh_jared -m -n
4. On robot PC: run $multi_waypoint_gps. Add GPS goals as required ... Verify robot moves to goal locations as desired. Record accuracy.
5. On base PC: run $./opstation_qt ... Current odometry and gps data should display to the screen, as well as log to file.

6.6 Obstacle Avoidance

1. Set up static obstacles
2. Put Segway in balance mode, start SICK scanner, start GPS unit
3. On robot PC(player/config): $player -d telemeter.so segwayrmp.cfg
4. On robot PC: run $vfh_jared -m -n
5. On robot PC: run $multi_waypoint_gps. Add gps goals as required ... Verify robot moves to goal locations as desired. Record accuracy.
6. Use people as dynamic obstacles as required
7. On base PC: run $./opstation_qt ... Current odometry and gps data should display to the screen, as well as log to file.

6.7 Cooperative Behaviour - Leader/Follower Test 1 (Waypoints)

1. Set up static obstacles
2. Put Segway #1 (mouse) and Segway #2 (cat) in balance mode, start SICK scanner, start GPS unit
3. On mouse robot PC(player/config): $player segwayrmp_vfh.cfg
4. On mouse PC: run $vfh_jared -m -n
5. On mouse PC: run $multi_waypoint_gps. Add gps goals as required for the mouse's route.
6. On cat PC: $player segwayrmp_vfh.cfg
7. On cat PC: $./vfh_jared -m -n
8. On base PC: run $./leader_follower
9. Use people as dynamic obstacles as required

6.8 Cooperative Behaviour - Leader/Follower Test 2 (Teleop)

1. Set up static obstacles
2. Put Segway #1 (mouse) and Segway #2 (cat) in balance mode, start SICK scanner, start GPS unit
3. On mouse robot PC(player/config): $player segwayrmp.cfg
4. On base PC: run $playerv -h 131.135.76.178

5. On base PC: In playerv: Devices->Position (0)->Subscribe, Devices->Position (0)->Command, Devices->Position (0)->Enable
6. On base PC: In playerv: Use interface to verify motion control of robot
7. On cat PC: $player segwayrmp_vfh.cfg
8. On cat PC: $./vfh_jared -m -n
9. On base PC: run $./leader_follower
10. Use people as dynamic obstacles as required

6.9 Cooperative Behaviour - Cat and Mouse

1. Set up static obstacles
2. Put Segway #1 (mouse) and Segway #2 (cat) in balance mode, start SICK scanners, start GPS units, radios
3. On mouse robot PC(player/config): $player segwayrmp_vfh.cfg
4. On mouse PC: run $vfh_jared -m -n
5. On mouse PC: run $multi_waypoint_gps. Add gps goals as required for the mouse's route.
6. On cat PC: $player segwayrmp_vfh.cfg
7. On cat PC, run $./vfh_jared -m -n
8. On base PC: run $./cat_and_mouse_client
9. Use people as dynamic obstacles as required

References

1. J. Borenstein and L. Feng. Umbmark - a method for measuring,comparing, and correcting dead-reckoning errors in mobile robots. *University of Michigan*, 1994.

2. The Segway Corporation. Segway robotic mobility platform. http://www.segway.com/segway/rmp/, 2004.

3. Brian Gerkey, Richard T. Vaughan, and Andrew Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. *Proceedings of the 11th International Conference on Advanced Robotics*, pages 317–323, 2003.

4. I. Ulrich and J. Borenstein. VFH+: Reliable obstacle avoidance for fast mobile robots. *IEEE Int. Conf. on Robotics and Automation*, 1998.

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for who the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in Section 8.)

   Defence R&D Canada – Suffield
   PO Box 4000, Station Main
   Medicine Hat, AB T1A 8K6

2. SECURITY CLASSIFICATION (overall security classification of the document, including special warning terms if applicable)

   UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title).

   Staged Experiments in Mobile Vehicle Autonomy (U)

4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)

   Giesbrecht, J. ; Collier, J. ; Monckton, S.

5. DATE OF PUBLICATION (month and year of publication of document)

   December 2004

6a. NO. OF PAGES (total containing information, include Annexes, Appendices, etc) 49

6b. NO. OF REFS (total cited in document) 4

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

   Technical Memorandum

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

   DRDC Suffield

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)

9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)

   DRDC Suffield TM 2004-288

10b. OTHER DOCUMENT NOs. (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

   (x) Unlimited distribution
   (   ) Distribution limited to defence departments and defence contractors; further distribution only as approved
   (   ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
   (   ) Distribution limited to government departments and agencies; further distribution only as approved
   (   ) Distribution limited to defence departments; further distribution only as approved
   (   ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally corresponded to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected).

   Unlimited

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C) or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

This paper presents details and results for experiments conducted at DRDC Suffield in autonomous robot control of the Segway RMP (Robotic Mobility Platform). The autonomous control was a demonstration of robotic abilities achievable with an inexpensive, rapid implementation by a small team using off-the-shelf hardware and software. It was based upon the Player/Stage robot software development environment, and used a basic sensor suite consisting of a SICK laser scanner, GPS localization, and robot distance encoders. The demonstration was able to achieve basic teleoperation, as well as autonomous dead-reckoning, GPS waypoint following, obstacle avoidance and leader/follower behaviour. Safety procedures and robot performance metrics applicable to all autonomous vehicle research were also developed over the course of this project, and are documented in this report.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifies, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

cooperative behaviour, autonomy, control, modularity, Segway